EXPRESS MAIL NO. EK673491681US Docket No. 00-8010

UNITED STATES PATENT APPLICATION

OF

Stuart J. JACOBS

Francis L. MANNIX, Jr.

Thomas W. CHRISTOFFEL and

Scott A. BELGARD

FOR

METHOD AND APPARATUS FOR SUPPORTING CRYPTOGRAPHIC-RELATED ACTIVITIES IN A PUBLIC KEY INFRASTRUCTURE

METHOD AND APPARATUS FOR SUPPORTING CRYPTOGRAPHIC-RELATED ACTIVITIES IN A PUBLIC KEY INFRASTRUCTURE

GOVERNMENT CONTRACT

The U.S. Government has a paid-up license in this invention and the right in limited circumstances to require the patent owner to license others on reasonable terms as provided for by the terms of Contract No. DAAL01-96-2-002 awarded by the U.S. Army.

FIELD OF THE INVENTION

The present invention relates generally to cryptography and, more particularly, to systems and methods for supporting cryptographic-related activities in a public key infrastructure.

BACKGROUND OF THE INVENTION

Public key cryptography has been commonly used to provide a mechanism to support access control and general authentication services in distributed environments. For example, in networks storing confidential information, conventional systems typically employ access control to limit access to the confidential information to designated parties. These systems may additionally employ general authentication services to authenticate users of various network resources to ensure that originators and recipients of messages are actually the parties they claim to be.

Conventional public key cryptography relies upon public key certificates, such as those defined in ITU X.509, to bind a user's public key reliably to his name and

provide users with the high level of assurance desired when identifying other entities. A certificate may be signed using a private key associated with the sender. The recipient of the message can then verify that the message was actually sent by the originator named in the message, provided that the recipient verifies the signature using the sender's public key.

In conventional systems, the responsibility for generating digital signatures and verifying digital signatures is borne by an application program executing on a general-purpose computer, under the control of a general-purpose operating system. For example, a conventional application program running on a computer in a network may have to authenticate link and network control messages. Similarly, the application program may have to encrypt messages using various encryption algorithms before transmitting these messages to other nodes in the network. Such tasks require an application programmer or network developer to acquire detailed knowledge of complex secret and public key algorithms and then develop programs to perform the required cryptographic functions.

Additionally, these cryptographic-related functions may require a network entity to exchange a number of messages with corresponding network entities when establishing a security association (SA). Such exchanges of messages consume very large amounts of network bandwidth, which is often limited in wireless systems.

As a result, there exists a need for a mechanism designed to avoid the requirement for an applications or network developer from having to acquire detailed knowledge of secret and public key algorithms. There is also a need for a mechanism

that integrates cryptographic-related functions into a simple to use service set, thereby simplifying the developer's task regarding inclusion of strong security features in application and infrastructure programs.

SUMMARY OF THE INVENTION

Systems and methods consistent with the present invention address these and other needs by integrating cryptographic-related functions in a software-based tool. The software-based tool includes a standardized interface that may be used by program developers for requesting the desired functions. The tool may also be customized to include any cryptographic-related functions based on the requirements of the particular user/system.

In accordance with the purpose of the invention as embodied and broadly described herein, a method for performing cryptographic-related functions in a network node is provided. The method includes receiving an input requiring cryptographic-related processing and generating a message based on the input. The message represents one of a predefined set of messages for processing by a cryptographic processing component. The method also includes transmitting the message to the cryptographic processing component and performing the cryptographic-related processing.

In another aspect of the present invention, a computer-readable medium, having sequences of instructions stored thereon is provided. The instructions may be invoked by a plurality of predefined messages and include sequences of instructions

which, when executed by a processor, cause the processor to receive an input representing one of the predefined messages. The instructions also cause the processor to transmit, based on the input, a request for cryptographic-related processing to a cryptographic processing module. The instructions further cause the processor to perform the cryptographic-related processing.

In still another aspect of the present invention, a cryptographic module is provided. The cryptographic module includes a memory configured to store a plurality of cryptographic processing programs where each program is invoked via one of a plurality of predefined messages. The cryptographic module also includes a processor configured to receive an input requiring cryptographic-related processing, generate one of the predefined messages based on the input, and transmit the message to a first one of the cryptographic processing programs. The processor is also configured to perform the cryptographic-related processing.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, explain the principles of the invention. In the drawings,

Fig. 1 is a block diagram of an exemplary system in which an implementation consistent with the present invention may be employed;

Fig. 2 is an exemplary block diagram of a node of Fig. 1 in which systems and methods consistent with the present invention may be employed;

Fig. 3 illustrates user space components and kernel space components consistent with an implementation of the present invention;

Fig. 4 is an exemplary diagram of kernel space components consistent with an implementation of the present invention; and

Fig. 5 is a flowchart of processing for providing cryptographic-related functions in a manner consistent with the present invention.

DETAILED DESCRIPTION

The following detailed description of the invention refers to the accompanying drawings. The same reference numbers in different drawings identify the same or similar elements. Also, the following detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims.

Systems and methods consistent with the present invention provide cryptographic-related functions in a software-based tool that may be used by an applications or network developer. The developer requests a particular function using a predefined set of messages transmitted to the software-based tool via a function call. The software-based tool then performs the desired cryptographic-related function.

EXEMPLARY SYSTEM CONFIGURATION

Fig. 1 is a diagram of an exemplary system 100 in which implementations of the present invention may be employed. The system 100 includes nodes 110, 120 and 130, server 140 and network 150.

Each of the nodes 110, 120 and 130 may include any type of computer device, such as a personal computer, a laptop, a personal digital assistant (PDA) or a similar device, with a connection to network 150. In an exemplary implementation of the present invention, the nodes 110-130 transmit/receive messages to/from other nodes over network 150 via wired, wireless, or optical connections. The network 150 may include the Internet, a local area network (LAN), wide area network (WAN), intranet or another type of network. Only three nodes are shown for simplicity. It should be understood, however, that any number of nodes may be included in system 100.

The server 140 may store certificates, public key information or other information required to verify/encrypt messages. For example, the server 140 may be a conventional light-weight directory access protocol (LDAP) server, an X.500 server or another type of server that stores certificates, certificate revocation lists (CRLs), or similar information. The nodes 110-130 may access the server 140 to retrieve various information needed to perform the authentication/verification functions, as described in more detail below.

EXEMPLARY NODE

Fig. 2 illustrates an exemplary node 110 of Fig. 1 in which methods and systems consistent with the present invention may be implemented. Node 110 includes a bus 210, a processor 220, a main memory 230, a read only memory (ROM) 240, a storage device 250, an input device 260, an output device 270, and a

communication interface 280. The bus 210 permits communication among the components of the node 110.

The processor 220 may include any type of conventional processor or microprocessor that interprets and executes instructions. Main memory 230 may be a random access memory (RAM) or another type of dynamic storage device that stores information and instructions for execution by processor 220. Main memory 230 may also store temporary variables or other intermediate information used during execution of instructions by processor 220. The ROM 240 may include a conventional ROM device or another type of static storage device that stores static information and instructions for processor 220. The storage device 250 may include any type of magnetic or optical recording medium and its corresponding drive, such as a magnetic disk or optical disk and its corresponding disk drive.

The input device 260 may include any conventional mechanism that permits an operator to input information to the node 110, such a keyboard, a mouse, a pen, voice recognition and/or biometric mechanisms, etc. The output device 270 may include any conventional mechanism that outputs information to the operator, including a display, a printer, a pair of speakers, etc. The communication interface 280 may include any transceiver-like mechanism that enables the node 110 to communicate with other devices and/or systems. For example, the communication interface 280 may include a modem or an Ethernet interface for communicating via a LAN. Alternatively, the communication interface 280 may include other mechanisms for communicating via a network, such as network 150.

Node 110, consistent with the present invention, performs cryptographic-related functions in response to processor 220 executing sequences of instructions contained in a computer readable medium, such as memory 230. A computer-readable medium may include one or more memory devices and/or carrier waves. Such instructions may be read into memory 230 from another computer-readable medium, such as a data storage device 250, or from a separate device via communication interface 280. Execution of the sequences of instructions contained in memory 230 causes processor 220 to perform the process steps that will be described hereafter. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the present invention. Thus, the present invention is not limited to any specific combination of hardware circuitry and software.

EXEMPLARY NODE COMPONENTS

Fig. 3 schematically illustrates the user space components and kernel space components in node 110, in an exemplary implementation consistent with the present invention. In the exemplary implementation, the user space components may reside in any one of or a combination of main memory 230, ROM 240 and storage device 250. In addition, the kernel space components may reside in any one of or a combination of main memory 230, ROM 240 and storage device 250. Additionally, both the user space components and kernel space components may interact with the other devices in node 110, such as processor 220.

The user space components and kernel space components are shown separated by a dotted line in Fig. 3. The user space components include user application program 310, public key authentication infrastructure (PKAI) control daemon 320, certificate database 330, PKAI operations daemon 340 and PKAI remote server daemon 350. The kernel space components include PKAI socket handler 360, PKAI call handler 370 and PKAI request handler 380.

The PKAI control daemon 320 initializes and shuts down PKAI services. A startup script may invoke the PKAI control daemon 320 with "start" and an optional pass phrase. The PKAI control daemon 320 may initialize the PKAI operations daemon 340 and the PKAI remote server daemon 350. In addition, the PKAI control daemon 320 shuts down the PKAI operations and remote server daemons 340 and 350 when the PKAI shuts down, such as when power to the node 110 is terminated.

The PKAI operations daemon 340 may communicate with PKAI request handler 380 via a user datagram protocol (UDP) socket. The PKAI operations daemon 340 performs local disk input/output on behalf of PKAI request handler 380. For example, the PKAI request handler 380 may use PKAI operations daemon 340 to store X.509 certificates to memory and retrieve X.509 digital certificates from memory, such as certificates database 330.

The PKAI remote server daemon 350 may also communicate with PKAI request handler via a UDP socket. The PKAI remote server daemon 350 retrieves information on behalf of PKAI request handler 380. For example, the PKAI remote server daemon 350 may retrieve X.509 digital certificates and certificate revocation

lists (CRLs) from a network accessible server, such as server 140 (Fig. 1). The PKAI remote server daemon 350 may store these certificates and CRLs in certificate database 330.

The PKAI system socket handler 360 may communicate with user application program 310 over a UDP socket. For example, the communication may include a request for cryptographic-related services, as discussed in more detail below. In this situation, the PKAI socket handler 360 generates a corresponding function call to the PKAI request handler 380 to perform the desired function. The PKAI call handler 370 may also receive system service calls from user application program 310 and PKAI control daemon 320. The PKAI call handler 370 then generates a corresponding function call to the PKAI request handler 380, based on the particular request.

Fig. 4 illustrates the PKAI components of Fig. 3 residing in the kernel space along with three cryptographic processing components, consistent with an exemplary implementation of the present invention. In an exemplary implementation, the cryptographic processing components are compiled into the kernel during a kernel re-build and their functionality invoked via a system service function call.

The kernel components illustrated in Fig. 4 include PKAI socket handler 360, PKAI call handler 370, PKAI request handler 380, PKAI RSA cryptoprocessing module 410, PKAI elliptic curve (EC) cryptoprocessing module 420 and PKAI keyed message digest algorithm 5 (MD5) cryptoprocessing module 430. Only three cryptoprocessing modules are shown for simplicity. Other cryptoprocessing modules

may be included in the kernel space based on the particular user/system requirements. Additionally, the details of the particular cryptoprocessing modules used in implementations of the present invention, such as modules 410-430, would be obvious to one of ordinary skill in this art and are not described herein.

Referring back to Fig. 3, a system service function call may be initiated by user application program 310 via the UDP socket to PKAI socket handler 360 or by a system service call directly to the PKAI call handler 370. The system service function call may also be initiated by PKAI control daemon 320 to PKAI call handler 370. Other methods of invoking the PKAI functions may also be used in alternative implementations. For example, the PKAI cryptographic-related functions may be invoked by any number of conventional call mechanisms based on the particular user/system requirements. In each case, the network or applications programmer need only be aware of the particular set of predefined messages needed to invoke the desired function. These messages are then incorporated into the user application program 310 as required.

As described above, the PKAI request handler 380 may receive requests for cryptographic-related services from a number of sources. In each situation, the PKAI request handler 380 receives the request and generates a function call to the appropriate cryptoprocessing module, such as one of cryptoprocessing modules 410-430. The details of performing the particular cryptographic-related functions are described in more detail below.

EXEMPLARY PROCESSING FOR PROVIDING

CRYTPOGRAPHIC-RELATED FUNCTIONS

Fig. 5 illustrates processing associated with performing cryptographic-related functions using the PKAI system. Processing begins with initialization of the PKAI system (step 510). The PKAI system may be initialized by a startup script that is executed after node 110 is powered up. After the PKAI system begins operating, the control daemon, operations daemon and remote server daemons 320, 340 and 350, respectively, operate as described with regard to Figs. 3 and 4.

Assume that the user application program 310 executes an instruction requiring cryptographic-related processing (step 520). Such an instruction may, for example, require verifying a digital signature transmitted with a certificate using an RSA, MD5, EC or digital signature standard (DSS) algorithm or generating an RSA, MD5, EC or DSS digital signature. The instruction may also require encrypting or decrypting data using an RSA, EC or other cryptographic algorithm. The instruction may further require retrieving a digital certificate or certificate revocation list from either the user space components or a remote server, such as server 140. The instruction may also include verifying a certificate's hierarchy, performing self-signed certificate processing, performing certificate age checking, or retrieving, verifying and storing a digital certificate in the node. In essence, the instruction may require performing any cryptographic-related function, based on the system requirements.

The user application program 310, after executing the instruction, generates a system service call to PKAI call handler 370 (step 530). The system service call,

consistent with the present invention, may be chosen from a predefined list of messages that are used to invoke PKAI services. For example, assume that the request is for verifying an RSA 512 bit digital signature transmitted with a certificate. In this case, the predefined message may be PKAI_RSA512ver. In this scenario, the user application program 310 may then transmit PKAI_RSA512ver to the PKAI call handler 370. The PKAI call handler 370 receives the request and forwards a corresponding function call to the PKAI request handler 380 (step 540).

In the example, the PKAI call handler 370 may transmit PKAI_RSA52ver_req to the PKAI request handler 380. The PKAI request handler 380 may then transmit the request message to the appropriate cryptoprocessing module for processing (step 540). In this example, the PKAI request handler 380 transmits PKA_RSA512ver_req to PKAI RSA cryptoprocessing module 410 (Fig. 4). The PKAI RSA cryptoprocessing module 410 then performs the desired function, i.e., verifies the status of the RSA 512 bit digital signature transmitted with the certificate (step 550). The PKAI RSA cryptoprocessing module 410 then transmits the result to the PKAI request handler 380 (step 550). After receiving the result, the PKAI request handler 380 forwards the result back to the user application program 310 that initiated the request (step 560). The result may optionally be transmitted to the user application program 310 via the PKAI call handler 370.

Systems and methods consistent with the present invention enable applications and network programmers to incorporate any required cryptographic-related processing by merely incorporating the desired call message. An advantage of the

invention is that the applications or network programmer is able to incorporate complex security features without having to gain detailed knowledge of complex secret and public key algorithms. Appendix A illustrates an exemplary set of PKAI function call messages that may be used in an implementation consistent with the present invention. It should be understood that additional function call messages may be used in alternative implementations consistent with the present invention.

The foregoing description of preferred embodiments of the present invention provides illustration and description, but is not intended to be exhaustive or to limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practice of the invention. For example, the PKAI system has been described as being resident in one of the network nodes that receives/transmits messages. In alternative implementations, the PKAI system may be located remotely from the network node. The scope of the invention is defined by the claims and their equivalents.